

Nom, Prénom : **SPECIMEN**  
Groupe :

Version française

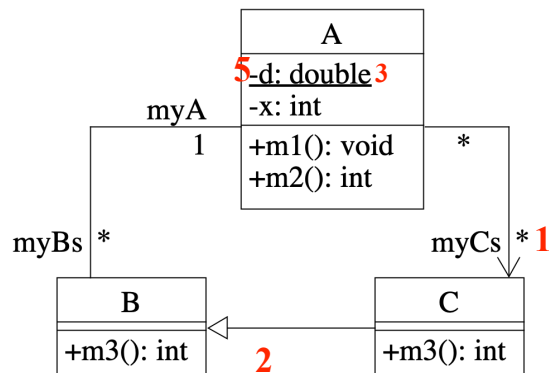
Répondre directement sur le sujet.  
**Examen factice**

## 1 Vocabulaire objet (5 pts)

Voici un code Java et le diagramme UML correspondant. Pour chaque élément présent et numéroté ci-dessous, notez le numéro de cet élément sur le diagramme UML **et** sur le code à un emplacement pertinent, **si** cet élément est bien présent (il peut être en plusieurs exemplaires, notez-le une fois dans le code et une fois dans le diagramme au maximum).

- une association vers une collection avec navigabilité à sens unique,
- un héritage,
- un attribut de classe,
- un lien d'usage, **X**
- une visibilité privée.

```
public class A {  
  5private 3static double d;  
  private int x;  
  private Set<B> myBs;  
  private Set<C> myCs;1  
  public void m1() {  
    x++;  
  }  
  public int m2() {  
    return x-(int)d;  
  }  
}  
public class B {  
  private A myA;  
  private B nextB;  
  int m3() {  
    return myA.m2()+1;  
  }  
}  
public class C extends 2 B {  
  @Override  
  int m3() {  
    return 0;  
  }  
}
```



## 2 Documentation de code (6 pts)

Ci-dessous, du code incomplètement rédigé et documenté pour la classe `SmartBulb`, permettant de représenter l'état d'une ampoule intelligente et modifier ses composantes couleur. Proposez un meilleur nom pour les identifiants listés, dites s'il s'agit de méthodes, variables, paramètres ou attributs, puis ajoutez, dans le code, la documentation manquante ( français ou anglais).

Identifiant	Nature	Nom proposé
M_C	Attribut	
r	Attribut	red (rouge)
to	Méthode	turnOff (eteindre)
pc	Paramètre	percent (pourcentage)

```
/**
 * SmartBulb class
 * @author (Not important)
 * @version 0.0
 */
public class SmartBulb {
    private final String URI;
    private final static M.C = 255;
    private int r, g, b;
    private int p;
    /**
     * Constructor for a smart bulb.
     * @param uri The bulb's URI (unique), makes it accessible online
     * @param power The bulb's power consumption in watts when fully lit
     */
    public SmartBulb(String uri, int power) {
        this.URI = uri;
        this.p = power;
    }
    /**
     * Method to set the bulb to full light (pure white, maximum power consumption)
     */
    public void fullLight() {
        r=g=b=M.C;
    }
    /**
     * Method to turn of the bulb entirely
     */
    public void to() {
        r=g=b=0;
    }
    /**
     * Method to set the bulb to a dim lighting (from fully off to fully lit)
     * @param pc: the percentage of light to use (0-100)
     */
    public void dl(int pc) {
        r=g=b=(M.C*pc)/100;
    }
    /**
     * Method to set the bulb to any arbitrary coloured light
     * @param red : the red component
     * @param green : the green component
     * @param blue : the blue component
     */
    public void srgb(int red, int green, int blue) {
        r = red; g = green; b = blue;
    }
    /**
     * Computes the current consumption of the bulb
     * @return the power (watts) consumed
     */
    public double cc() {
        return ((r+b+c)*p)/(M.C*3.0);
    }
}
```

### 3 Étude de cas (9 pts)

#### 3.1 Sujet

On étudie un logiciel de *payeur de laverie automatique*. Ce logiciel permet de commander un distributeur de lessive, des lave-linge et des sèche-linge. Il dispose d'un panneau d'entrée (numérique) permettant de choisir chacun de ces éléments, d'un afficheur qui affiche la somme restant à insérer, et d'une fente pour introduire des pièces (une à une). Il y a également un bouton d'annulation (qui permet de restituer la somme introduite). Le distributeur de lessive agit immédiatement, mais les autres appareils sont indisponibles quand ils sont utilisés et envoient un signal électronique au logiciel quand ils deviennent de nouveau disponibles ; le logiciel doit refuser qu'un appareil soit sélectionné s'il n'est pas disponible (et afficher un message de refus).

#### 3.2 Contexte

Complétez :

Acteurs externes			
<b>:USER</b>	<b>:WASHING_POWDER_DISPENSER</b>		<b>:DRYER</b>
	<b>:WASHING_MACHINE</b>		

Événements externes			
<b>:USER</b>	<b>:WASHING_MACHINE</b>		<b>:DRYER</b>
sélectionne un élément		signale que la machine est disponible	
introduit une pièce			
appuie sur « annuler »			

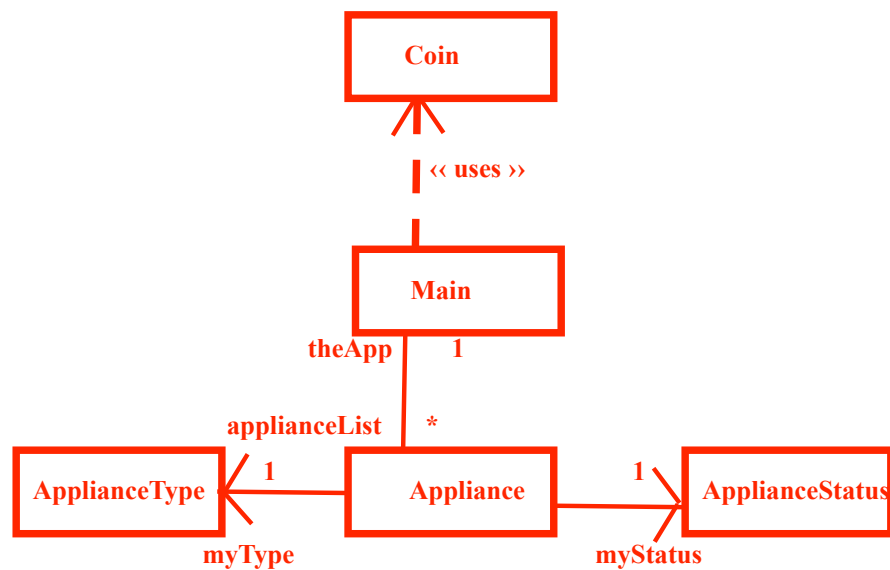
Événements résultats			
<b>:USER</b>	<b>:WASHING_POWDER_DISPENSER</b>		
	<b>:WASHING_MACHINE</b>		
	<b>:DRYER</b>		
afficher message de refus		envoyer instruction de mise en route	
afficher somme restante			
afficher confirmation paiement			
afficher message d'annulation			

### 3.3 Diagramme de classes

Quelles sont les entités qui se dégagent dans ce logiciel ?

**Au moins de quoi stocker les machines cibles disponibles, leur type, et leur état ; éventuellement les pièces de monnaie**

Proposez un diagramme de classes en spécification générale.



**Indispensable : classe principale, classe pour les appareils**

**Les valeurs des pièces sont utilisées dans le paiement (alternative : stocker toutes les pièces déjà introduites, ce qui permet de rentre la monnaie).**

**(Les classes Type et Status peuvent être remplacées par des valeurs entières ou booléennes.)**